

## **REMARKS**

Claim 1 calls for initializing the video stream from a video server when a first application requests video. The office action makes it clear that the asserted first application is the plug-in.

The claim further calls for providing the video stream for the first application. Again, the office action makes it clear that the video is provided for the first application.

Next, the claim calls for monitoring to detect if the first application crashes while receiving the video stream. It is unclear what element in the cited reference meets this limitation. There is nothing that can be found in any of the cited materials which in any way relates to monitoring to detect if the application crashes. The office action does indicate, in paragraph 2, that if the plug-in crashes, access is maintained to the video stream as connection data saved in order to be made available to the next created instance of the plug-in. However, it is respectfully suggested that this does not indicate that crashing of the plug-in is monitored. To the contrary, if the argument set forth in paragraph 2 were accepted as true, there is still no monitoring for a crash. According to the argument set forth in paragraph 2, access is maintained whether or not the plug-in crashes or not, without any monitoring. There is no discussion of monitoring in any of the cited material.

The claim also calls for, if the first application crashes, maintaining access to the video stream for a second application through the video server. This does not appear to be the case in the reference. Since access to the video stream, according to the argument set forth in paragraph 1 of the office action, is established by the plug-in, once the plug-in crashes, it is not seen how video could continue to be accessed. Since it is the plug-in that is providing the video, it seems impossible that the video stream could be continued to be accessed after its crash.

Thus, for both reasons, it is respectfully requested that the rejection of claim 1 should be reconsidered. First, there is no monitoring in the cited reference and, second, there is no maintaining access after the plug-in crashes.

Therefore, reconsideration of the rejection of claims 1-20 is respectfully requested.

Claim 28 calls for initializing a video stream using a video server in response to a request for a video from a first application. If the first application crashes, access to the video stream is maintained for a second application through the video server and directing the server to release the video stack. Claim 28 was rejected as obvious over Semenzato in view of Bopardikar.

Again, the argument is maintained that access to the video stream is maintained even if the plug-in crashes. But since it is the plug-in which obtains the video and audio, it is not seen how access could be maintained.

Further, it is contended that Bopardikar teaches directing the server to release the video stack. Cited in support thereof is column 5, lines 22-49. That material relates to storing of HDTV frames on a disk drive. There is no discussion of directing a server to release the video stack, a video stack, or what happens when a first application crashes.

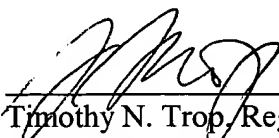
Also, cited is column 26, line 23 through column 27, line 25. This material talks about the situation when a disk failure takes place. However, a disk failure has nothing to do with an application crashing, maintaining access to a video stream, or directing a server to release a video stack. Thus, a disk failure has nothing to do with an application failure. Moreover, there is no suggestion of releasing a video stack since no video stack would be implemented. The system simply stores data on a disk. In a case of a failure of the disk, the data continues to be read out. It is explained in column 27, that since full frames of data are transferred to the router, the video environment is unaware of the disk failure. In such case, it would certainly be surprising that a video stack were released by the server.

Therefore, it is respectfully submitted that Bopardikar has nothing to do with the claimed invention, it relates to a non-analogous art of storing data on disks. It does not have the situation where an application crashes. It does not maintain access to the video stream for a second application through any video server. It does not direct the server to release a video stack since the server is not involved in a disk operation.

Therefore, it is respectfully submitted that claim 28 and its dependent claims should be in condition for allowance.

Respectfully submitted,

Date: August 20, 2004

  
\_\_\_\_\_  
Timothy N. Trop, Reg. No. 28,994  
TROP, PRUNER & HU, P.C.  
8554 Katy Freeway, Ste. 100  
Houston, TX 77024  
713/468-8880 [Phone]  
713/468-8883 [Fax]